PCT/SE 99 / 02169

PRV

PATENT- OCH REGISTRERINGSVERKET
Patentavdelningen

4

REC'D 14 FEB 2000
WIPO PCT

Intyg
Certificate

SE99/2169

#6

Härmed intygas att bifogade kopior överensstämmer med de handlingar som ursprungligen ingivits till Patent- och registreringsverket i nedannämnda ansökan.

This is to certify that the annexed is a true copy of the documents as originally filed with the Patent- and Registration Office in connection with the following patent application.

(71) Sökande Applicant (s) *Net Insight AB, Stockholm SE*

(21) Patentansökningsnummer Patent application number *9804023-1*

(86) Ingivningsdatum Date of filing *1998-11-24*

Stockholm, 2000-02-07

För Patent- och registreringsverket
For the Patent- and Registration Office

Emma Johnsson

Emma Johnsson

Avgift
Fee

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN COMPLIANCE WITH RULE 17.1(a) OR (b)

PATENT- OCH REGISTRERINGSVERKET SWEDEN
Postadress/Adress Box 5055 S-102 42 STOCKHOLM
Telefon/Phone +46 8 782 25 00 Vx 08-782 25 00
Telex 17978 PATOREG S
Telefax +46 8 666 02 86 08-666 02 86

METHOD AND SYSTEM FOR DETERMINING NETWORK TOPOLOGY

Technical Field of Invention

The present invention refers to a method and a system for determining the topology of a network of nodes that are interconnected via unidirectional connections.

Background of the Invention

A communication network is a data processing system that includes a plurality of interconnected components, or nodes, such as work stations, phones, data storage devices, printers, servers, switches, routers, hubs, etc. The nodes are typically interconnected via unidirectional connections, for example in the form of optical fibers, and communicate by transmitting and receiving messages to and from other nodes on said unidirectional connections.

In order for a node to know how a selected destination is reached, i.e. in which direction to send a message destined for an intended receiver, there is a need for each node to know the topology, also referred to as architecture or configuration, of the network, or at least of a portion thereof.

One way of providing each node with information on the network topology is to use a centralized scheme in which a central source will provide a map of the network to all other nodes of the network, for example as described in US 5,654,958 (Natarajan). A disadvantage of this solution is that each change in the network topology has to be brought to the attention of the central source and has to be addressed by the central source if said change is to come to the other nodes attention, automatically adding signaling overhead between the central source and the network nodes. Also, if the central source is down, updating of network topology is temporarily rendered impossible.

Another way of providing each node with information on the network topology is to use a distributed scheme in

which messages, containing information on the local network topology, are exchanged between the nodes of the network. Based upon received topology messages, each node will generate and maintain its own map of the network, or at least of a local portion thereof. An example of such a solution has been described in US 5,682,479 (Newhall et al.), wherein each network node is arranged to transmit vector-routed packets cross the network in various specified direction, each packet gathering information about the network topology along its way. The packets are then returned to the originating node with the gathered information.

As another example, US 5,506,838 (Flanagan) describes a prior art solution wherein so called discovery packets are from one or more source nodes to other nodes of the network, thereby informing the network nodes on network topology.

Objects of the Invention

A disadvantage with the latter two examples on prior art is that the schemes regard the network in terms of bi-directional connections, each typically comprising two unidirectional connections. If, for example, some point-to-point-connection between two nodes in the network lacks bi-directional connectivity, part of the network topology may be impossible to determine. Furthermore, computation of routing tables and the like may result in suboptimized solution.

An object of the invention is therefore to provide a distributed scheme for determining network topology.

Another object is to provide a scheme that does not have to rely solely on bi-directional point-to-point connectivity.

Yet another object is to provide a scheme wherein the amount of messages transmitted within the network in order to determine the network topology is kept low.

Summary of the Invention

The above mentioned and other objects of the invention are achieved by a method and a system according to the accompanying claims.

The invention thus provides a method and a system for determining the topology of a network of nodes that are interconnected via unidirectional connections. According to the invention, the existence of a network loop within said network is determined using message forwarding among the nodes of the network. Information of the existence of said network loop is then distributed to nodes within said network.

The invention is thus based upon the idea of regarding the network in terms of network loops (at least when determining network topology), and to determine and distribute information on the existence of such loops.

According to a preferred embodiment, a node of the network will transmit a message, sometimes referred to below as a topology discovery message, from an output port and will subsequently determine reception of a forwarded version of said message at an input port, thereby indicating the existence of a network loop. Furthermore, each node of the network that receives a topology discovery message is preferably arranged to forward said message on at least one, typically all, of its output ports.

Preferably, each or at least a plurality of the nodes of the network will be arranged to transmit and detect messages of this kind. Furthermore, preferably all nodes of the network will be arranged to forward such messages.

Having determined the existence of a loop, information referring thereto is distributed to nodes of the network, typically nodes forming at least part of said network loop and/or other nodes of the network, i.e. nodes that do not form part of said loop, and will typically include information as to which nodes, and which

ports thereof, that form part of said network loop. Preferably, generation and distribution of such information is also performed using message forwarding.

Ideally, transmission and forwarding of one single and comparatively small message will be enough to determine the existence of a network loop. Advantageously, such a loop will be determined even though the network, and more specifically the loop as such, may comprise a number of point-to-point connections that lack bi-directional connectivity, which in some cases would have been impossible in prior art.

In comparison, the above mentioned US 5,506,838 describes a message forwarding scheme wherein each node receiving a so called discovery packet will determine whether the information contained therein has been previously received. If the information is new, then the receiving node records the packet information, modifies the packet to identify itself, and forwards the packet to other nodes. However, if the information has been previously received, propagation of the packet is terminated to prevent unnecessary packet transmissions. Consequently, in this prior art, when receiving a packet that contains information that has been previously received, instead of determining the existence of a network loop and distributing information referring to said network loop to nodes forming part of the loop, as suggested by the invention, no use is made of the gained information except for simply preventing further forwarding of said message.

According to a preferred embodiment, a node having two or more outgoing ports, and having not yet been able to determine which one of said output ports that is part of a specific network loop, is arranged to transmit two or more respective messages from respective output ports, each message identifying the respective output port used for transmission thereof and thereby enabling subsequent determination of which one of said two or more output

ports of said node that forms part of said network loop. For example, the sending of such two or more different messages may be initiated by the reception of a topology discovery message or may be initiated by the sending node itself. These messages may then be used subsequently, for example at the sending node or at another node forming part of the network loop having received said message, or a forwarded version thereof. The advantage of this scheme is of course that it eliminates uncertainties as to which ports that form part of which loops in using an efficient and simple mechanism.

Expressed in terms of steps performed by nodes of the network, a specific example of the aforementioned embodiment comprises the steps of: transmitting a message from an output port of a first node; receiving said message, as such or in a forwarded version, at an input port of a second node; transmitting two or more modified versions of said message from respective two or more output ports of said second node, each modified version identifying the respective output port used for transmission thereof; receiving one of said modified versions of said message at said first node, thereby identifying which one of said two or more output ports that forms part of said network loop; and transmitting a message from said first node to said second node, said message identifying the output port of said second node that forms part of said network loop.

Information as to which nodes, and preferably also which ports thereof, that form part of a determined network is generated in a preferable manner using message forwarding. Typically, each node receiving a topology discovery message, or correspondingly, will forward said message and, at the same time, include information as to the identity of the forwarding node, and typically also of the output port thereof, into said message. The loop information generated in such a manner may then be dis-

tributed to the nodes forming said loop, preferably also using message forwarding.

Concluding, the solution of regarding the network in terms of network loops, determining the existence of such loops, and distributing information on such loops at least to nodes that form part of said loops, clearly forms an inventive idea involving an inventive step.

The above mentioned and other aspects, features and details of the invention will be more fully understood from the following description of a preferred embodiment thereof.

Brief Description of the Drawings

An exemplifying preferred embodiment of the invention will now be described in detail with reference to the accompanying drawings, wherein:

Figs. 1a and 1b schematically show respective network topologies;

Fig. 2 schematically shows a flow chart of a topology discovery algorithm according to a preferred embodiment of the invention: and

Figs. 3-7 schematically illustrate an exemplifying network and the exchange of messages between nodes of the network according to the exemplifying topology discovery algorithm shown in Fig. 2.

Detailed Description of a Preferred Embodiment

In Fig. 1a, a simple network is shown, comprising three nodes 1, 2, and 3 interconnected via unidirectional connections 4, 5, 6, and 7. As indicated by semi-circular dotted arrows, connections 4 and 7 form a network loop comprising nodes 1 and 2, and connection 5 and 6 form another network loop comprising nodes 2 and 3. Moreover, even though not explicitly indicated in Fig. 1a, connections 4, 5, 6, and 7 may also be said to form an overall network loop, i.e. the loop from node 1 via connection 4 to node 2, via connection 5 to node 3, via connection 6

to node 7, and via connection 7 back to node 1. Even though such a loop may be defined, the description below will primarily be focused on the discovery of the smallest possible network loops that each node is part of. Also, as is seen in Fig. 1a, each one of the indicated network loops actually form a bi-directional connection connecting the respective nodes.

In Fig. 1b, another simple network is shown comprising four nodes 10, 20, 30, and 40 interconnected via unidirectional connections to form a ring topology. In this case, the ring topology forms a network loop that comprises all four nodes, as indicated by the semi-circular dotted arrow in the center of the figure.

A flow chart of an embodiment of a topology discovery algorithm according to a preferred embodiment of the invention will now be described with reference to Fig. 2. As is understood, the main object of this topology discovery algorithm is to determine the existence of network loops of the kind indicated in Figs. 1a and 1b and to provide corresponding information to the nodes that form part of the respective network loop.

With reference to Fig. 2, the topology discovery algorithm comprises a loop detection step S10, a master announce step S20, a split-point reduction step S30, a loop list build-up step S40, a loop list distribution step S50, and a route table computation step S60.

Each of the steps of Fig. 2 will now be described with reference to an exemplifying network illustrated in Figs. 3-7, said network comprising six nodes 10, 20, 30, 40, 50, and 60. The node 10 has two output ports 11 and 13 and two input ports 12 and 14. Also, each one of the remaining six nodes is provided with two output and two input ports that are denoted correspondingly.

As shown in figs. 3-7, output port 11 of node 10 is connected via a unidirectional connection to input port 22 of node 20, output port 21 of node 20 is connected via a unidirectional connection to input port 32 of node 30,

output port 31 of node 30 is connected via a unidirectional connection to input port 42 of node 40, and output port 41 of node 40 is connected via a unidirectional connection to input port 12 of node 10. Furthermore, output port 43 of node 40 is connected via a unidirectional connection to input port 52 of node 50, output port 51 of node 50 is connected via a unidirectional connection to input port 62 of node 60, and output port 61 of node 60 is connected via a unidirectional connection to input port 44 of node 40. In this example, it is assumed that remaining input and output ports are unconnected.

An example of a loop detection step S10 of the topology discovery algorithm in Fig. 2 according to the preferred embodiment of the invention will now be described with reference to Fig. 3. During the loop detection step S10, the nodes of the network will transmit so called probe messages that are used to detect the presence of loops in the network topology. A node transmits probe messages on all output ports that, are not part of already determined loops. When a node generates probe messages, each message is provided with a unique identification identifying the probe message origin, i.e. identifying the output port that the message is transmitted from, for example the unique MAC address of the output port. The nodes of the network are arranged to forward received probe messages on all output ports. When forwarding a probe message, the content of the received probe message is mapped into the transmitted message. In other words, the content of the forwarded probe message will essentially be a copy of the content of the received probe message, thus identifying output port of the node that originated the probe message. The distribution of probe messages is limited by a hop-count mechanism that limits the number of hops that a probe message is forwarded over. (For simplicity, in the illustrated example, it is assumed that the number of hops that a message is allowed to travel is set to four.)

When a node receives one of its own probe messages from another node, it will determine that a loop exists, and it will know which input port and output port that forms part of this new loop. The node then becomes a so-called build-up master for the new loop and continues to the master announce step for the new loop.

In the example shown in Fig. 3, node 10 transmits a probe message PR(11) on its output port 11 to node 20, said probe message identifying the origin of the probe message. Node 20 then forwards the probe message on output port 21 to node 30, having incremented the hop-count indicated in the probe message by one. Node 30 forwards the probe message on output port 31 to node 40. Since node 40 has two output ports, it forwards the probe message on output port 41 to node 10 as well as on output port 43 to node 50. Node 50 will then forward the probe message on output port 51 to node 60. Since the maximum number of allowed hops has been reached, node 60 will decide not to forward the probe message. However, at the same time, node 10 will have received its own probe message from node 40 on input port 12 and will therefore determine that a network loop exists and that output port 11 and input port 12 are part of this new loop. Node 10 will then take on the role as build-up master and continue to the master announce step.

The master announce step S20 of the topology discovery algorithm according to a preferred embodiment of the invention as shown in Fig. 3 will now be described with reference to Fig. 4. When a node has determined the presence of a new loop using probe messages as described with reference to Fig. 3, it will take on the role as build-up master for the new loop. In order to let other nodes learn about the existence of the loop, the build-up master will send out a so-called master announce message on the output port previously identified as being part of the new loop.

The master announce message is forwarded by the same rules as the probe messages and serves two purposes. The first purpose is to inform other nodes about the existence of a new loop and to assign an identifier to the new loop, typically being the MAC address as mentioned above. This loop identifier is contained in all subsequent messages concerning the new loop and allows several loops to be discovered simultaneously without risking mix up of messages referring to different new loops. Upon receiving the master announce message, the other nodes become so called build-up slaves for the new loop and automatically know which of its input ports that are part of the loop. The second purpose of the master announce message is that it provides a mechanism for build-up master arbitration. If two nodes simultaneously receive their own probe messages, they will both try to take on the role as build-up master. In this preferred embodiment, this is resolved by a precedence mechanism based on the MAC addresses of the build-up masters. If a build-up master receives a master announce message (on the input port for which it is currently trying to become build-up master) from a node with a higher MAC address, it retreats, at least temporarily, and becomes build-up slave instead. If a build-up master receives a master announce message from a node with a lower MAC address, the master announce message is not forwarded.

When the build-up master eventually receives its own master announce message, it knows that all other nodes in the new loop have become build-up slaves and that it is the only build-up master for the new loop. The build-up master then continues to the split-point reduction step.

In the example shown in Fig. 4, node 10, being the build-up master, transmits a master announce message MA(11) on its output port 11 to node 20, said master announce message identifying the origin of the message, thereby also forming a unique loop identifier. This message is then forwarded in a similar manner as the probe

message described with reference to Fig. 3. Since node 40 has two output ports, it will forward the master announce message on both output ports. As each one of nodes 20, 30, 40, 50, and 60 receives the master announce message MA(11), it will take on the role as build-up slave. Correspondingly, when node 10 receives its own master announce message from node 40 on input port 12, it will know that all other nodes in the network loop have become build-up slaves, and will then move on to the split point reduction step.

An example of the split-point reduction step S30 of the topology discovery algorithm according to the preferred embodiment of the invention will now be described with reference to Figs. 5a-5c. A so-called split-point node is a node that has two or more connected output ports. When a node forwards a probe message or a master announce message, it must forward the messages on all its output ports, since it does not know which port that is part of the new loop. The goal of the split-point reduction step is to determine which one of the output ports of the split-point node that forms part of the new loop.

The split-point reduction step is started by the build-up master, which will send out a so called split-point announce message on the output port where it previously sent out the probe and master announce messages. The split-point announce message is provided with an identifier of the output port that it was sent on.

Split-point announce messages are forwarded according to the following rule: If the forwarding node has only one single output port, or if it already knows which output port that is part of the new loop (i.e. it has already been "resolved" as discussed below), the split-point announce message is forwarded in an unmodified version via the correct (or only) output port. Otherwise, the node is a considered a split-point node. It then sends out its own new split-point announce messages on all its output ports, instead of the received split-point

announce message. Moreover, each new split-point announce message contains an identifier of the output port that it was sent on.

When the build-up master receives a split-point announce message with an identifier for an output port of another node, it knows that that output port is part of the new loop. The build-up master informs the split-point node about this by transmitting a so-called split-point reduce message identifying that output port. Split-point reduce messages are forwarded in the same way as probe and master announce messages. When a split-point node receives a split-point reduce message identifying one of its output ports, it knows that that port is part of the new loop. The split-point has now been resolved, and the split-point reduce message need not be forwarded. Consequently, when the next split-point announce message is received at the node, it is forwarded unmodified on the now determined output port for the new loop.

Moreover, when a split-point node has been informed about which of its output ports that is part of the new loop, it sends out a so called release branch message on all other output ports. This is done to inform build-up slaves downstream from those ports that they are not part of the loop and they can now remove all protocol state regarding the loop and try to establish other loops instead.

Directly after sending out a split-point reduce message, the build-up master sends out a new split-point announce message to find the next split-point node. The split-point reduction step thus continues by reducing one split-point at a time starting from the split-point closest to the input port of the build-up master and working its way back to the output port of the build-up master.

When the build-up master finally receives one of its own split-point announce messages, it knows that there are no more split-points in the loop. All the nodes

forming part of the loop now knows which of its ports that are part of the loop, and all messages regarding the new loop is therefore transmitted only to the nodes involved in the loop. So far, however, each node only has information about its own ports, i.e. no node has complete knowledge of all nodes in the new loop (except in very simple topologies). The build-up master therefore continues to the so-called loop list buildup step.

In the example shown in Figs. 5a-5c, the build-up master node 10 starts by sending out a split-point announce message SPA(11) on the output port 11 to node 20. The split-point announce message SPA(11) is provided with an identifier of the output port 11 that it was sent on. The message SPA(11) is forwarded by nodes 20 and 30 to node 40. However, as node 40 has two connected output ports, node 40 sends out its own new split-point announce messages SPA(41) and SPA(43) on its respective output ports, each message identifying the output port that it was sent on. When the build-up master node 10 receives the message SPA(43) from node 40 in Fig. 5a, it knows that the identified output port 41 is part of the new loop. As shown in Fig. 5b, the master node 10 therefore informs the split-point node 40 about this by transmitting a split-point reduce message SPR(41) identifying the output port 41, said message being transmitted/forwarded the same way as the previous messages. When node 40 receives the split-point reduce message SPR(41) identifying its output port 41, it knows that port 41 is part of the new loop. Node 40 then sends out a release branch message RB on the remaining output port 43. As node 50 and 60 then receive the release branch message RB, they cease to be build-up slaves and may start searching for other loops. As shown in Fig. 5c, directly after sending the split-point reduce message SPR(41), the build-up master 10 sends out a new split-point announce message SPA(11). As the split point at node 40 has now been resolved, the new split point announce message SPR(11)

will be forwarded all the way back to the master node 10. The master node 10 will therefore determine that there are no more split-points in the loop.

An example of the loop list build-up step of the topology discovery algorithm according to the preferred embodiment of the invention will now be described with reference to Fig. 6. During the loop list build-up phase, the build-up master collects information about which nodes, and ports thereof, that are part of the new loop using message forwarding. The build-up master initiates the loop list build-up step by sending out an empty loop list (LB in Fig. 6) on the output port for the new loop. Each build-up slave on the loop path to the build-up master's input port adds itself to the loop list and forwards the new loop list to the next node.. When the loop list reaches the build-up master again, the build-up master adds itself to the end of the loop list. The loop list build-up phase is now finished and the build-up master has complete knowledge of the topology of the loop. It can then move on to the loop list distribution step.

An example of the loop list distribution step of the topology discovery algorithm according to the preferred embodiment of the invention will now be described with reference to Fig. 7. During this step, the build-up master informs all the nodes forming part of the new loop about the topology of the loop using message forwarding. The same message format is used to distribute the list as was used during the loop list build-up phase. The build-up master transmits the list on the output port where the new loop has been established. Each build-up slave node that receives a loop list under distribution must forward the list to the output port belonging to the same loop that the list arrived on. The build-up master that has originated the loop list must make sure that the list comes back via the input port belonging to the same loop as the output port that the list was originated on. If the loop list does not arrive within a configured time

interval, or if an error is detected by a node during the loop list distribution, the list is re-originated.

When the build-up master has received the full and correct loop list as transmitted, it will cease to operate as build-up master, consider the new loop to be up and the loop state fully built. Correspondingly, when a build-up slave has received the full and correct loop list, it will cease to operate as a build-up slave, consider the loop to be up and the loop state fully built. The (newly ceased) build-up master and slaves then continue to the routing table computation step.

In the routing table computation step, when a node has received and accepted a new loop list, it will use it to compute an updated route table based on the available loops considered valid. The route table will contain one item for each reachable node. Each item typically contains the output port that must be used to reach the destination and the MAC address of the input interface of the destination.

As understood by those skilled in the art, the above mentioned steps may be altered, modified, and/or integrated. Furthermore, steps may be added or excluded based upon the desired functionality within the scope of the invention, which is defined by the accompanying claims.

For example, according to an alternative embodiment, the loop detection step, the split-point-reduction step, and the loop list build-up step is integrated into one single step, wherein the rules for handling a loop detection message will include the split-point and loop list build-up features. An advantage of such a scheme is that it limits the amount of messages transmitted between the nodes of the network. On the other hand, it increases message size and processing.

As another example, other ways of limiting message forwarding than by using a simple hop-count mechanism may be selected.

Based upon the inventive idea, many diffenet topology message handling rules may be used, for example deterimining how and when a node is to become loop master, how and when to send probe messages, how and when to look for new loops, and so on, the scope of the invention of couse not being limited to the specific embodiment described in detail above.

Hence, the decision regarding how to actually realize and implement the invention will typically depend upon how the explicit network type will be positively or negatively affected by aspects such as the amount of messages transmitted within the network, message size, the amount of message processing, the changing and/or maintaining of states at each node, and so on.

CLAIMS

1. A method for determining the topology of a network of nodes that are interconnected via unidirectional connections, said method comprising the steps of:

determining the existence of a network loop within said network using message forwarding among said nodes; and

distributing information on the existence of said network loop within said network.

2. A method as claimed in claim 1, wherein said step of determining the existence of a network loop comprises the steps of:

transmitting a message from an output port of a first node; and

receiving a forwarded version of said message at an input port of said first node.

3. A method as claimed in claims 1 or 2, wherein said step of determining the existence of a network loop comprises the steps of receiving a message at an input port of a node and forwarding said message on one or more output ports thereof.

4. A method as claimed in any one of the preceding claims, wherein said step of distributing information on the existence of said network loop comprises using message forwarding for distributing said information.

5. A method as claimed in any one of the preceding claims, wherein said information comprises information as to which nodes that form part of said network loop.

6. A method as claimed in any one of the preceding claims, wherein said information comprises information as to which ports that form part of said network loop.

7. A method as claimed in any one of the preceding claims, including the steps of:

transmitting two or more messages from respective two or more output ports of a node, each message identifying the respective output port used for transmission thereof; and

receiving a message referring to one of said two or more messages, thereby identifying which one of said two or more output ports of said node that forms part of said network loop.

8. A method as claimed in any one of the preceding claims, including the steps of:

transmitting a message from an output port of a first node;

receiving said message, as such or in a forwarded version, at an input port of a second node;

transmitting two or more modified versions of said message from respective two or more output ports of said second node, each modified version identifying the respective output port used for transmission thereof;

receiving one of said modified versions of said message at said first node, thereby identifying which one of said two or more output ports that forms part of said network loop; and

transmitting a message from said first node to said second node, said message identifying the output port of said second node that forms part of said network loop.

9. A method as claimed in any one of the preceding claims, wherein forwarding a message comprises the step of including information as to the identity of the forwarding node into said message.

11. A method as claimed in any one of the preceding claims, wherein forwarding a message comprises the step

of including information as to the identity of the output port that said message is transmitted from into said message.

12. A system for determining the topology of a network of nodes that are interconnected via unidirectional connections, comprising a first node that is arranged to transmit a message from an output port thereof, to determine the existence of a network loop within said network by determining reception of a forwarded version of said message at an input port thereof, and, as a result, to distribute information on the existence of said network loop to nodes within said network.

13. A system as claimed in claim 12, comprising one or more second nodes being arranged to forward said message on one or more output ports thereof when receiving said message on an input port thereof.

14. A system as claimed in claims 12 or 13, wherein said nodes are arranged to distribute said information on the existence of said network loop by message forwarding.

15. A system as claimed in any one of claims 12-14, wherein a node that has two or more outgoing ports is arranged to transmit two or more respective messages from respective output ports, each message identifying the respective output port used for transmission thereof and thereby enabling subsequent determination of which one of said two or more output ports of said node that forms part of said network loop.

16. A system as claimed in any one of claims 12-15, comprising:

a first node transmitting a first message;

a second node receiving said message, as such or in a forwarded version, and transmitting two or more

modified versions of said message from respective two or more output ports, each modified version identifying the respective output port used for transmission thereof,

wherein said first node is arranged to identify which one of said two or more output ports, of said second node, that forms part of said network loop by determining reception of one of said modified versions of said message at said first node, and, as a result, to transmit a message from said first node to said second node, said message identifying the output port of said second node that forms part of said network loop.

ABSTRACT

The present invention relates to a method and a system for determining the topology of a network of nodes that are interconnected via unidirectional connections.

According to the invention, the existence of a network loop within said network is determined using message forwarding among said nodes, and, as a result, information on the existence of said network loop is distributed to nodes within said network.

Elected for publication: Fig. 1b

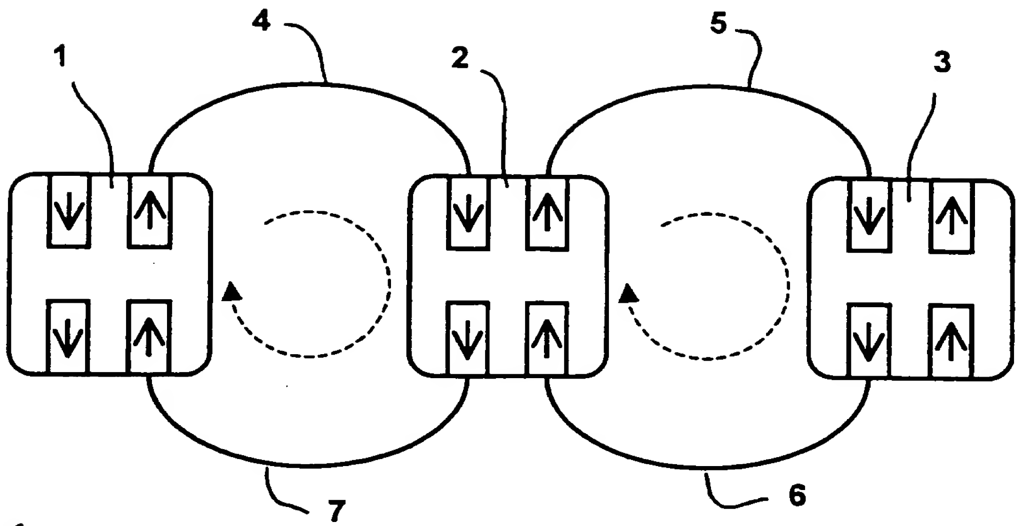


Fig. 1a

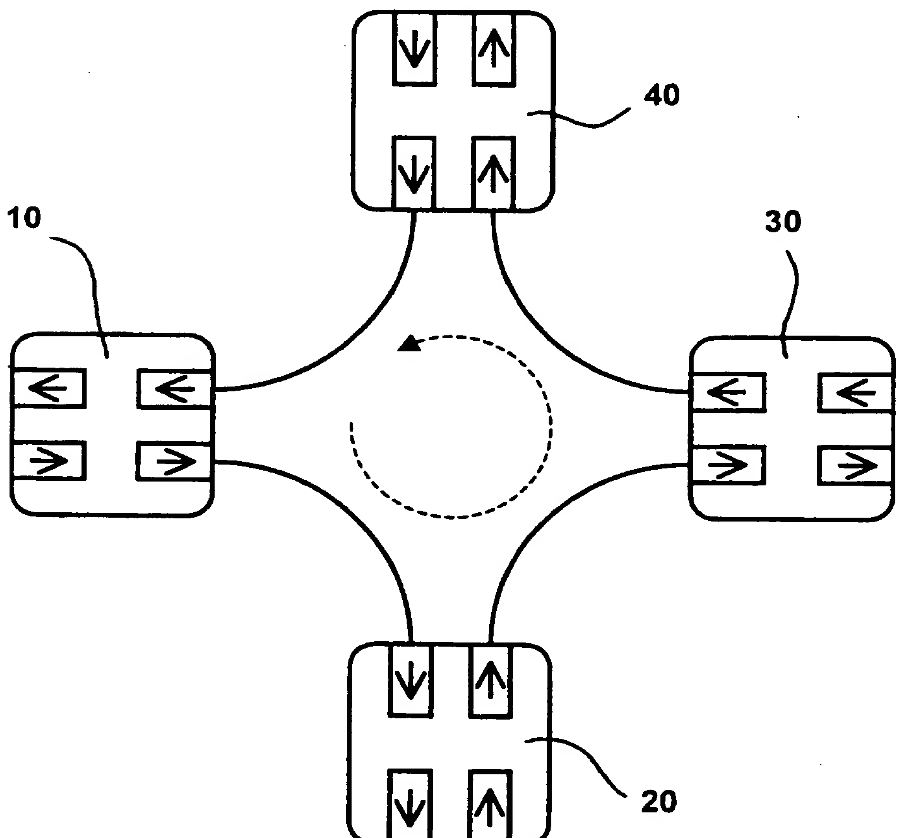


Fig. 1b

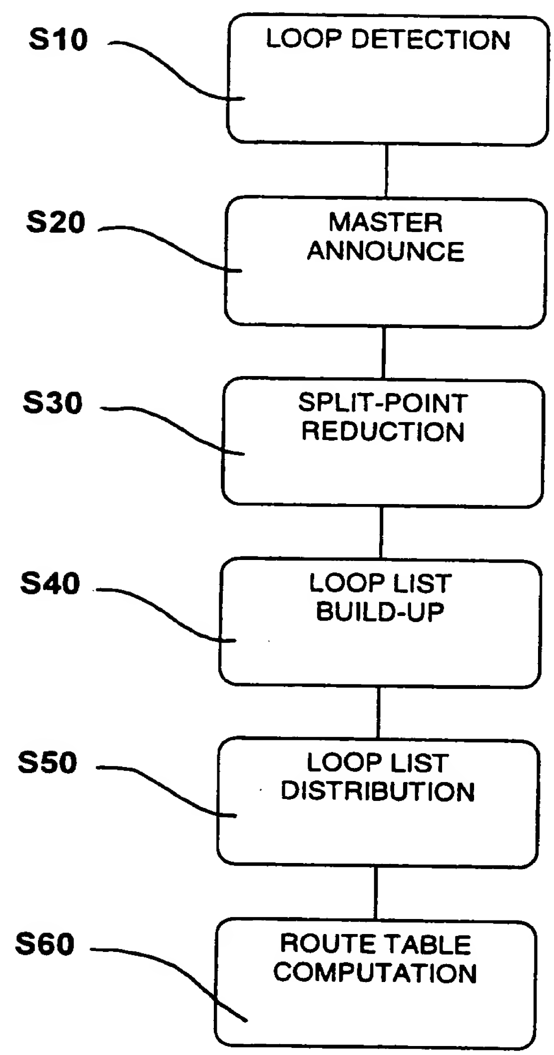

S10
LOOP DETECTION
S20
MASTER ANNOUNCE
S30
SPLIT-POINT REDUCTION
S40
LOOP LIST BUILD-UP
S50
LOOP LIST DISTRIBUTION
S60
ROUTE TABLE COMPUTATION

Fig. 2

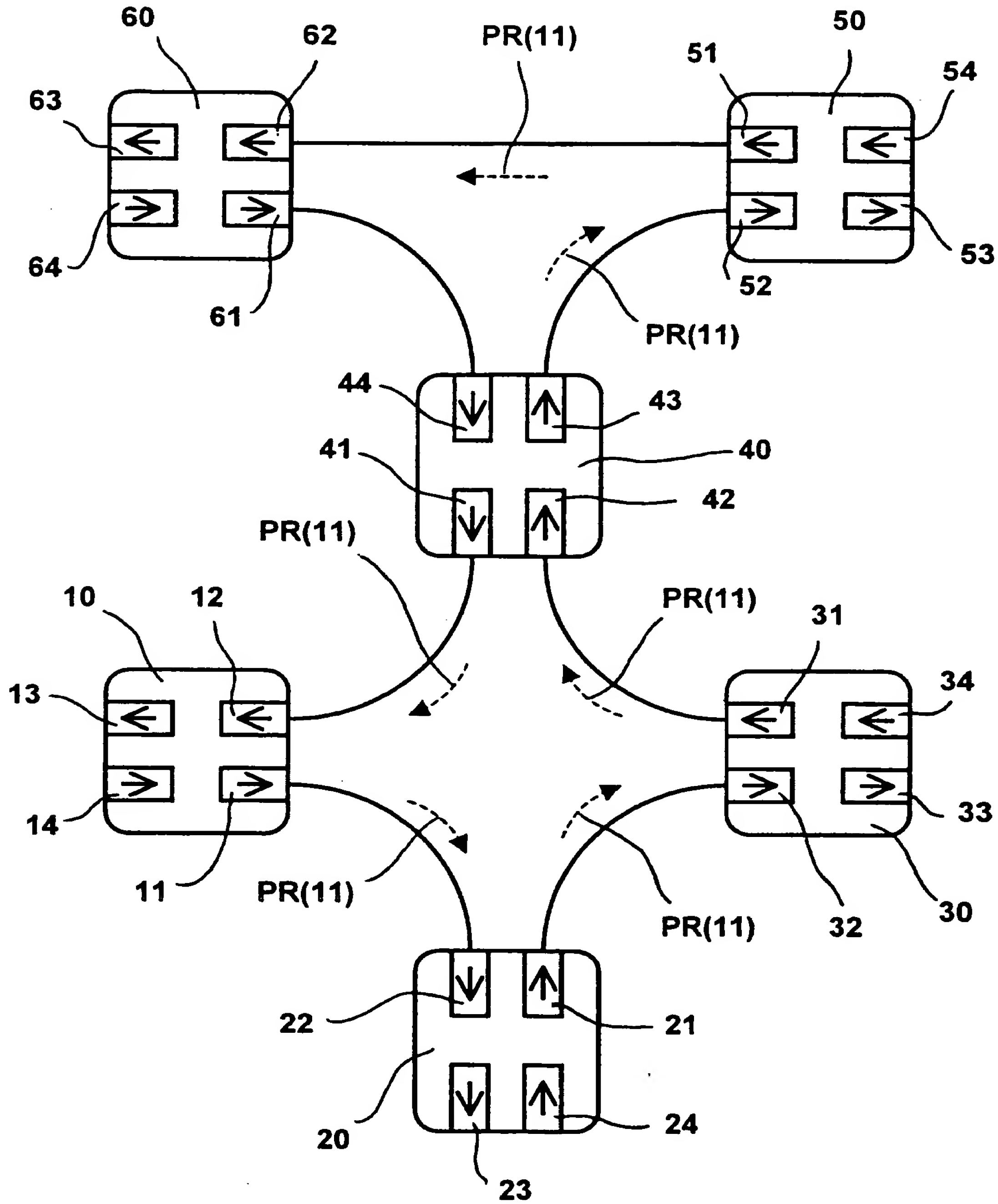

60
62
PR(11)
50
51
54
63
64
61
53
52
PR(11)
44
43
41
40
42
PR(11)
10
12
PR(11)
31
13
34
14
33
11
PR(11)
PR(11)
32
30
22
21
20
24
23

Fig. 3

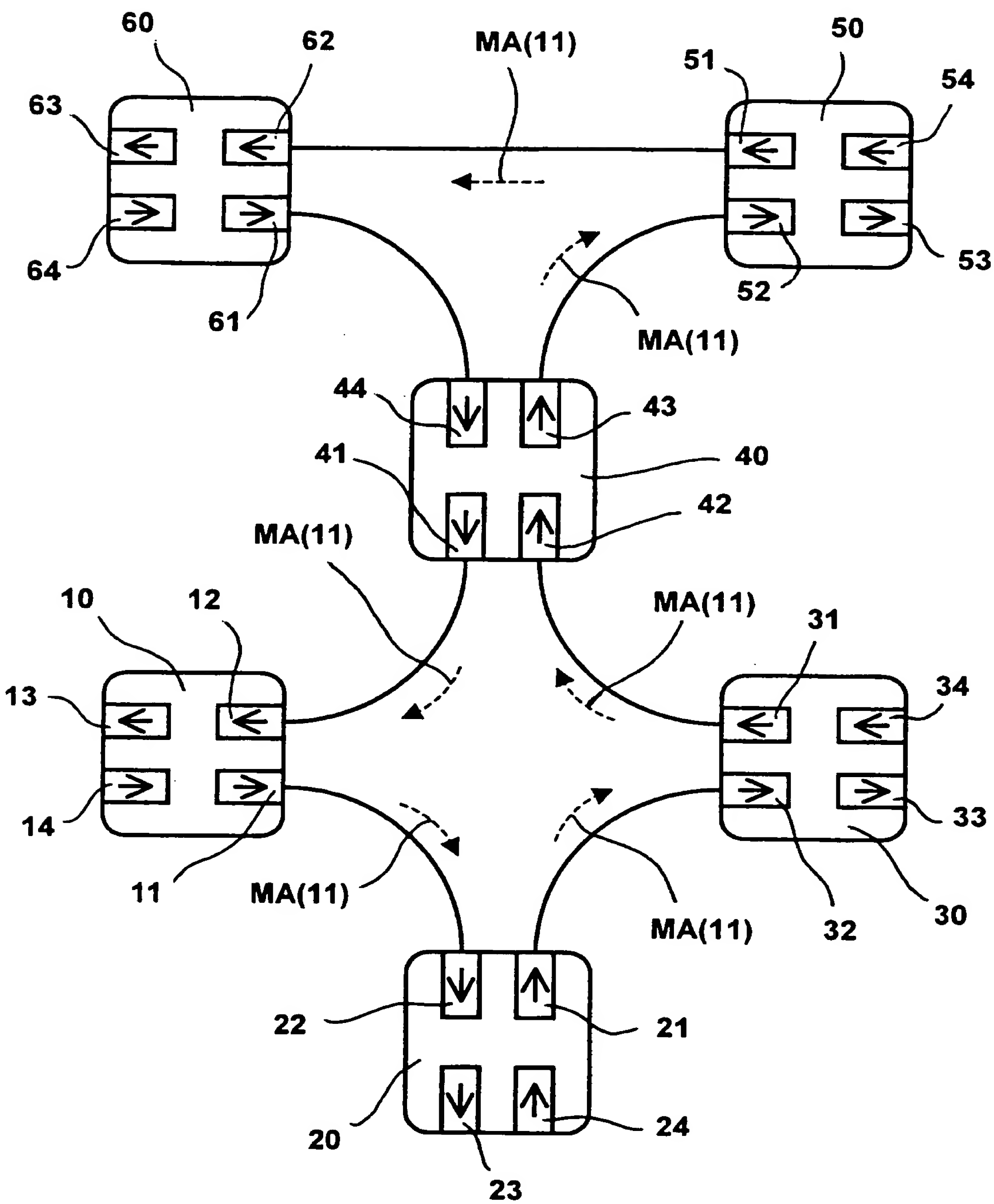
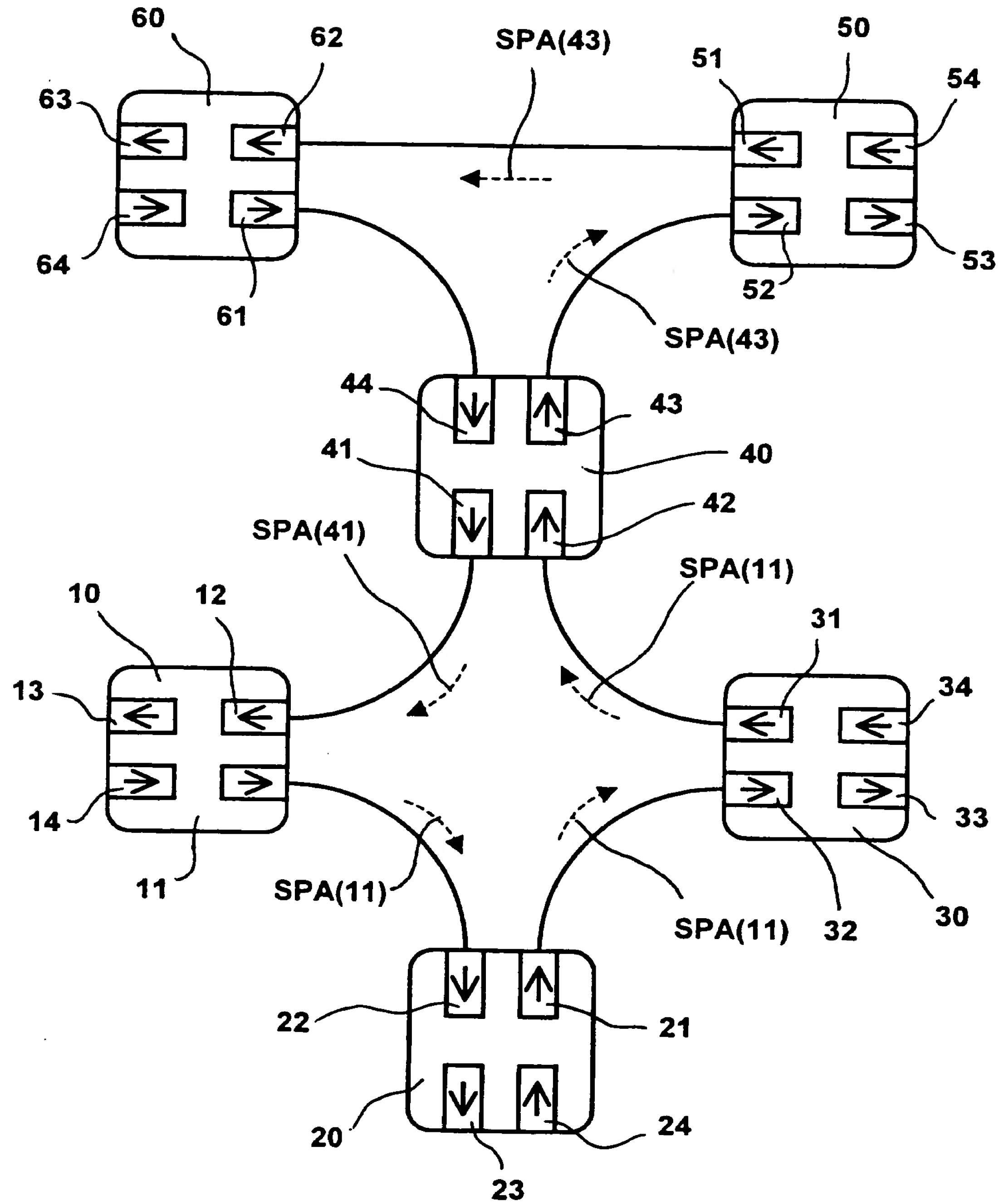
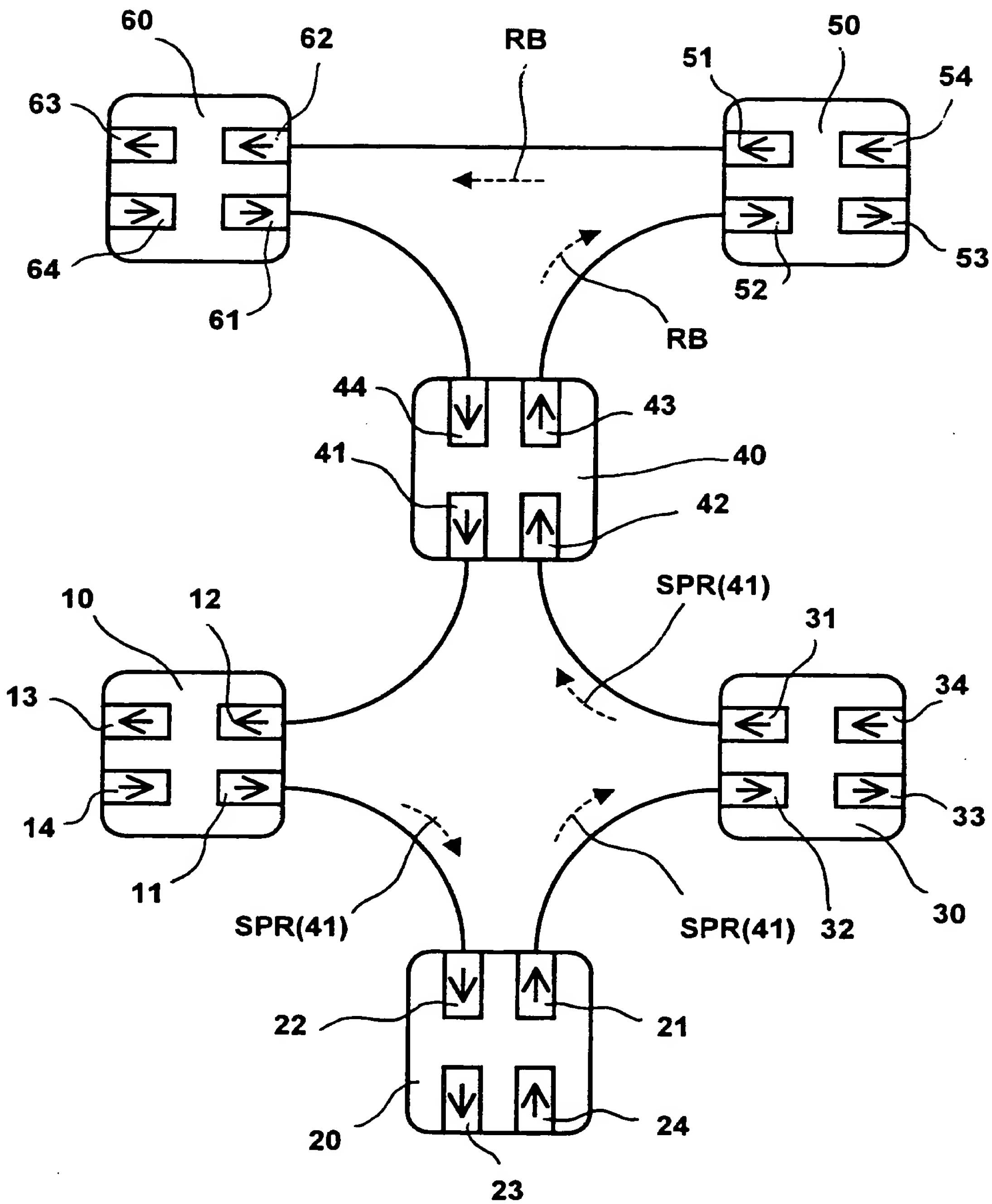

60
62
MA(11)
50
51
54
63
64
61
52
53
MA(11)
44
43
41
40
42
MA(11)
10
12
MA(11)
31
13
34
14
33
11
MA(11)
MA(11)
32
30
22
21
20
24
23

Fig. 4



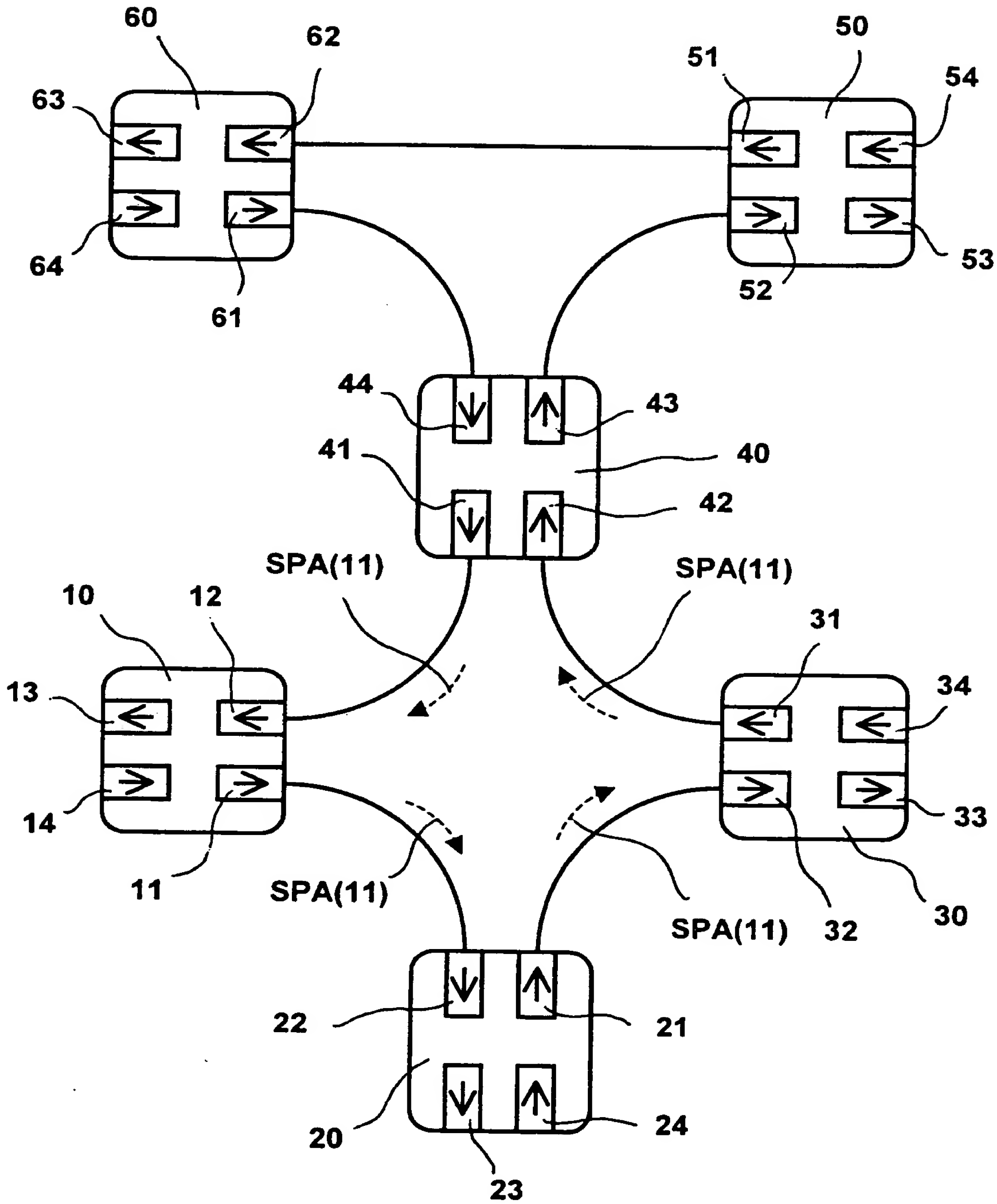
60
62
SPA(43)
50
51
54
63
64
61
52
53
SPA(43)
44
43
41
40
42
SPA(41)
SPA(11)
10
12
31
13
34
14
11
SPA(11)
SPA(11)
32
33
30
22
21
20
24
23

Fig. 5a



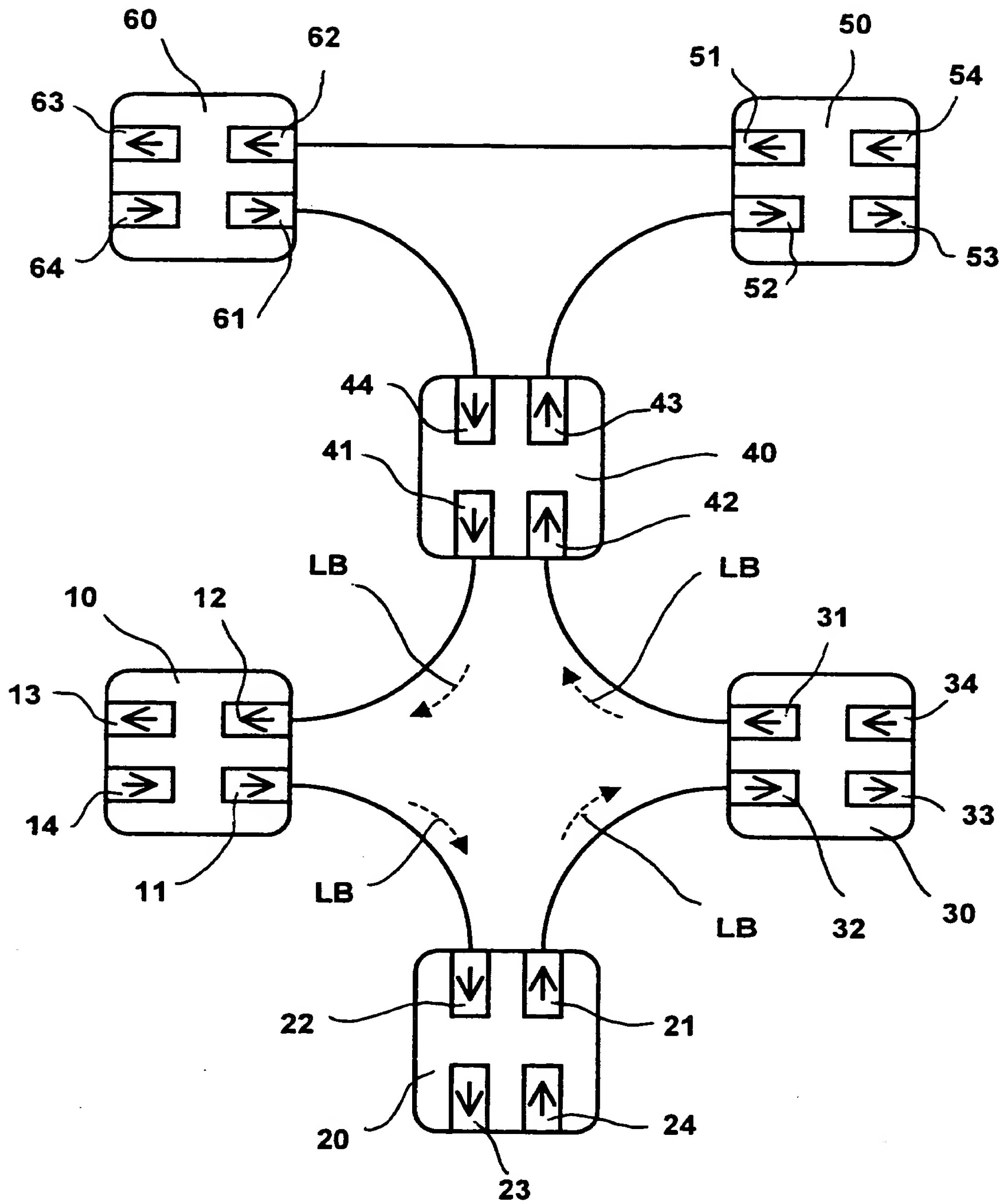
60
62
RB
50
51
54
63
64
61
52
53
RB
44
43
41
40
42
SPR(41)
10
12
31
13
34
14
11
33
SPR(41)
SPR(41)
32
30
22
21
20
24
23

Fig. 5b



60
62
63
64
61
50
51
54
53
52
44
43
41
40
42
SPA(11)
SPA(11)
10
12
13
14
11
31
34
33
32
30
SPA(11)
SPA(11)
22
21
20
24
23

Fig. 5c



60
62
63
64
61
50
51
54
53
52
44
43
41
40
42
LB
LB
10
12
13
14
11
31
34
33
32
30
LB
LB
22
21
20
24
23

Fig. 6

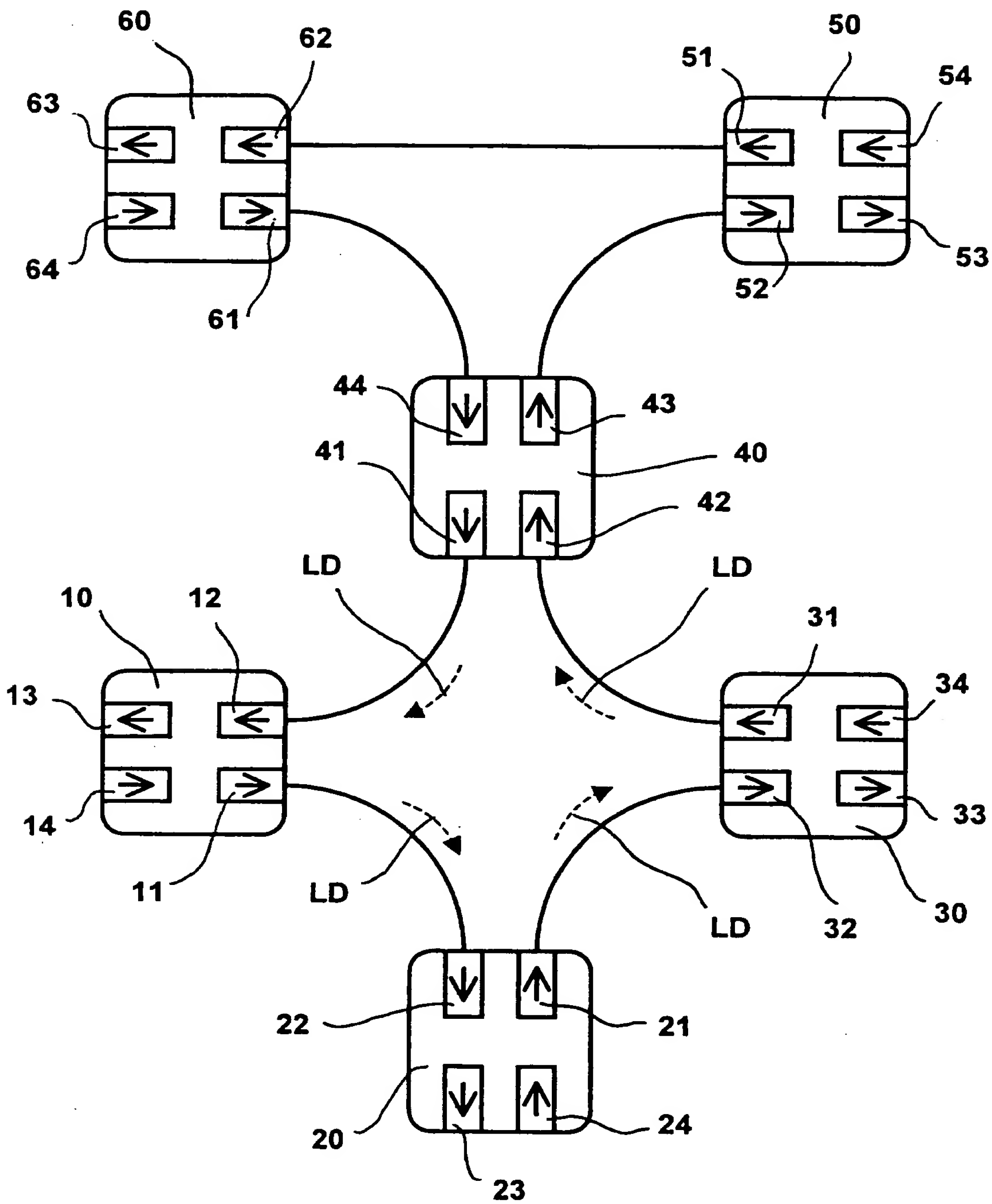

60
62
63
64
61
50
51
54
52
53
44
43
41
40
42
LD
LD
10
12
13
14
11
31
34
32
33
30
LD
LD
22
21
20
23
24

Fig. 7